All-Optical Tree-based Greedy Router

Sahel Sahhaf, Abhishek Dixit, Wouter Tavernier, Didier Colle, Mario Pickavet, Piet Demeester
Ghent University – iMinds, Department of Information Technology (INTEC)
sahel.sahhaf@intec.ugent.be

Abstract: Forwarding logic in greedy routing systems requires less memory and fewer components than longest-prefix match-based forwarding in IP routing. We demonstrate an all-optical design of a greedy router with desirable scalability and energy-efficiency characteristics enabling high data rate throughput.
OCIS codes: (230.1150) All-optical devices; (200.4660) Optical logic.

1. Introduction

Greedy routing stems from the idea of geographic routing in which every node in a network receives a GPS coordinate [1]. Network nodes can route greedy by relaying incoming packets to the neighbors that are closer to the packet's intended destination. By repeatedly applying this distance-decreasing policy, the destination can be reached. The same idea can be reused by assigning virtual coordinates to the network nodes. The greedy routing scheme requires nodes to only store the coordinates of their neighbors. This makes it more memory-efficient than traditional IP routers based on longest prefix matching, taming excessive growth in forwarding/routing tables.

Moreover, routers consume significant amounts of energy. As optical technologies have an excellent potential to be included in high capacity and energy-efficient routing systems, we design a greedy router using optical components. The designed optical router enables high data rates, is more energy-efficient and more scalable in terms of the routing table size compared with conventional IP routers. We demonstrate a tree-based greedy router implemented through interconnection of SOAs and couplers. The functionality of the proposed circuit is verified through simulation.

2. Tree-based greedy routing

In greedy routing, greedy embeddings are used to avoid situations in which packets are stuck in local minima. This ensures that there is always a neighboring router which brings a packet closer to the destination than any other neighbor. In [2], we proposed a greedy embedding based on a spanning tree of the network which assigns coordinates to the nodes reflecting its path from the root of the tree to the node. In this scheme, first a rooted spanning tree of the network is generated, and all coordinates of the root node are set to zero. Then the children of each node are numbered from 1 to d. Finally, each node calculates the coordinates of its children by adding the number assigned to each child after the last non-zero coordinate in its own coordinate set (CS). Fig. 1 shows an example for this embedding.

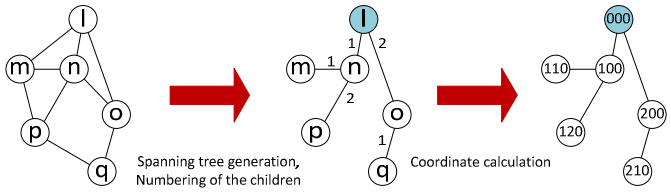

Fig.1 Steps for network nodes numbering

Once the nodes CSes are calculated, packets can be forwarded towards neighbors which guarantee to decrease the distance towards the destination. In this context, tree distance is used as the metric. This is the hop count on the tree between two nodes. We illustrate tree distance calculation between nodes (1,1,0) and (1,2,0) in Fig. 1. The first common coordinates in the CSes of the two nodes shows the CS of the closest common ancestor which is (1,0,0). The number of non-zero coordinates in each CS after the common coordinates determines the number of hop counts to the common ancestor which is one for both CSes in the example. Therefore, the tree-distance (sum of the hop counts) between them is '2'.

3. Greedy router architecture

We assume that the CSes are assigned to the network nodes and every node knows the CSes of its neighbors. The major functionality of a greedy router is based on 2 components: i) a distance calculator, which calculates the tree-distance between every neighbor and the destination of a packet and ii) a comparator, which compares the calculated tree-distances and finds the minimum among them (next hop). In this paper, we propose a novel circuit for tree-distance calculation and the second component is based on existing all-optical comparators [3]. The architecture for tree-distance calculation for one neighbor is demonstrated. The proposed distance calculation functionality is

required for every neighbor of a given greedy router (multiple components can calculate distance in parallel for different neighbors). Fig. 2 depicts the circuit of the proposed tree-distance calculator. The main components of this circuit are XOR, OR, AND gates, a counter and a flip flop. The general idea in this circuit is to check the given two CSes and find the first common coordinates and start counting the non-zero coordinates after that in both sets.

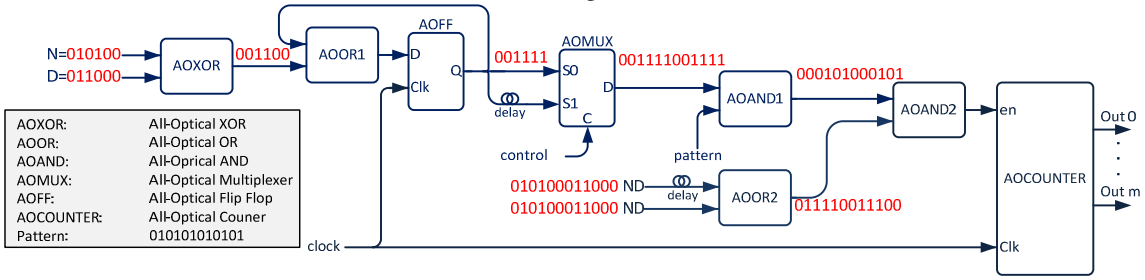

Fig. 2 All-optical tree-distance calculator for one neighbor

In Fig. 2, the CSes of the destination (D) and the neighbor (N) are applied to the inputs of the AOXOR gate. These inputs are sequences of bits. The XOR gate determines the uncommon bits in the two CSes. Using an AOOR1 and a feedback from AOFF, a signal can be obtained which is '1' from the moment the first uncommon bit in the two CSes is detected. Using a multiplexer and a fiber delay element, we generate the output of the AOFF two times consecutively, as it is needed for each CS. The duration of the fiber delay is equal to the length of a CS. This signal indicates the start for counting the coordinates in each set. In order to count the coordinates and not the bits, a pattern is used which determines the end of each coordinate in a set by '1'. The output of the AOAND1 is '1' only if the first uncommon bit is detected and it is also the end of a coordinate. Next, it is required that only non-zero coordinates in each set are counted.

In the second part of the circuit, the two CSes are fed into AOOR2 as sequences of bits consecutively (ND). One of them goes through a fiber delay element with the length of one bit. Assuming that a coordinate consists of two bits, if there is at least one bit equal to '1' in each coordinate of a set, the last bit of the corresponding coordinate would be '1' in the output of the AOOR2. Fig. 2 depicts the circuit where each coordinate consists of 2 bits. In case of 'N'-bit coordinates, 'N-1' single-bit fiber delay elements and an N-way AOOR are required.

Finally, as the output of AOAND2, a signal is produced which is '1' only at the end of each coordinate if the first uncommon coordinate in the two CSes is detected and the coordinate is also non-zero. Applying this signal to the enable pin of a counter, the hop count of both nodes to their closest common ancestor is counted. In Fig.2 the output at different stages of the architecture is marked for a sample input. The counter can be based on existing all-optical counters [4]. For other functionalities of an all-optical router such as clock recovery and packet header separation, we refer to previous works [5].

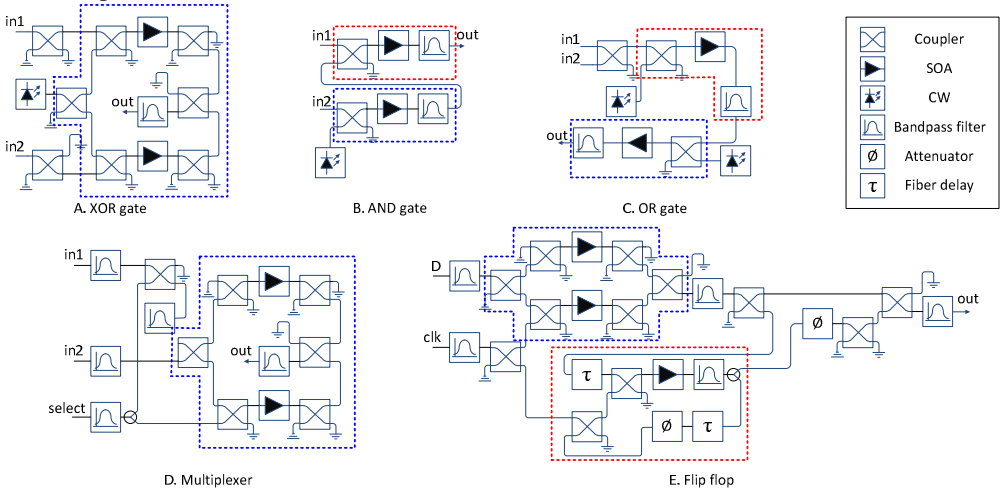

Fig. 3 All-optical components design

4. All-optical components

The all-optical design of the components used in the greedy router architecture are illustrated and explained briefly in this section. For a XOR gate, an SOA-MZI configuration (Fig. 3.A - dashed line area) is used. The two inputs are fed to the arms of SOA-MZI. In case of '1' at the inputs, due to the phase shifts through the couplers, the output

signal would be destructive and it would be '1' only if one of the inputs is '1'. The emulation of an AND gate is through 2 inverters (Fig. 3.B - dashed line area). In the emulation of the OR gate, the 2 inverters (Fig. 3.C) are used to create only 2-level signals. The design of a multiplexer is also based on a SOA-MZI configuration working as a bar and cross state switch. The emulated flip flop is level-triggered. The SOA-MZI configuration selected on top of the circuit (Fig. 3.E) puts the input data to the output in case the clock is '1'. In case there is no clock, the selected part in the bottom works as a feedback loop which maintains the output without any change.

5. Experimental results

In order to evaluate the functionality of the proposed circuit for tree-distance calculation, we simulate the schematic in Fig. 2 in the VPI software suite. In this simulation, we consider wavelength sources with a center frequency of 193.12 THz (λ1) and 201.12 THz (λ2). We assume ND (input in Fig. 2) sources at λ1 and all the other sources are assumed at λ2. The continuous wave (CW) light sources, within the gates are chosen appropriately. The clock signal has a data rate of 20Gbps.

The all-optical components are tested with a data rate of 10Gbps. As illustrated in Section 4, the logic gates require 2 SOAs, and the multiplexer and the flip flop consist of 3 SOAs. For the delay elements, a simple fiber-based delay element is used. The fiber length is such that it induces a delay equivalent to the duration of a data bit.

In this experiment, we chose the CSes of 3 coordinates and each coordinate of 2 bits, resulting into a total size of 6 bits. Fig. 4 depicts the inputs for the destination and neighbor CSes (D and N, (1,2,0) and (1,1,0) respectively). The final output of the simulated circuit is depicted in Fig. 4. Applying this output to the enable pin of a counter results into calculating the tree-distance between them. Having the rise/fall time of roughly 5 picoseconds in the output signal, the proposed design is capable of functioning appropriately in high data rates of up to 100 Gbps.

In the following, we analyze the power consumption of the proposed design. As couplers are passive components, SOAs dominate the total power consumption of the proposed optical greedy router. Based on the required number of SOAs, we roughly estimate the power consumption of the circuit.

Knowing that the proposed design scales with the degree of the nodes, in an Autonomous System (AS) border router with 64 peers, there are 64 distance-calculators and one 64-way comparator/63 2-input comparators for each input port. The coordinates should have enough bits to be able to number all the peers of an AS. Therefore, 6 bits are considered in each coordinate. Based on these assumptions and considering a 5-bit counter in each tree-distance calculator (assuming the maximum depth of 16 in the tree), the total number of SOAs is 2874. Assuming the power consumption of 100mW for each SOA (independent of the data rate), the proposed circuit for one port in the router consumes 287.4W. Although, this power estimation is only based on the number of SOAs, it gives a general overview on the energy-efficiency of the circuit compared to a Juniper T-series 100Gbps slot card with power consumption of 542W. Note that the estimated values are based on a discrete design. Applying photonic integration technology will significantly improve the energy-efficiency of the router.

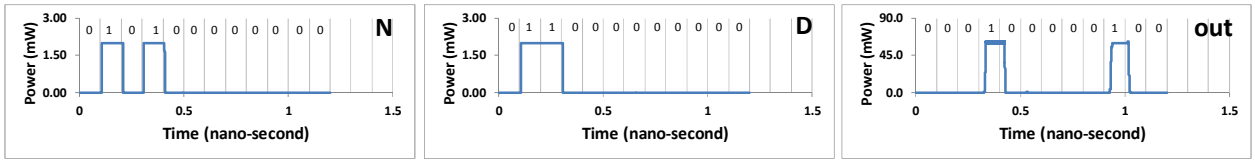

Fig. 4 Inputs for neighbor (N) and destination (D) coordinate sets and the output of the simulated circuit

6. Conclusions

For the first time, to the best knowledge of the authors, an all-optical tree-based greedy router was demonstrated. The circuit scales with the degree of nodes in the network and is more memory-efficient than longest-prefix match-based IP routers. The proposed design was constructed through interconnection of SOAs and couplers and implemented for a sample input (6-bit coordinate sets). Simulation results confirmed the feasibility of the proposed design for high data rates up to 100Gbps. A power estimation based on the number of SOAs suggests improved energy-efficiency of the design compared to IP routers.

7. References

[1] B. Karp et al., "GPSR: greedy perimeter stateless routing for wireless networks," in Mobile computing and networking. ACM, 2000.
[2] S. Sahhaf, et al., "Single failure resiliency in greedy routing," in DRCN, 2013.
[3] M. Scaffardi et al., "All-optical comparator based on cross gain modulation in semiconductor optical amplifiers," in OFC, 2008.
[4] J. Wang et al., "All-optical counter based on optical flip-flop and optical AND gate," in ECOC, 2009.
[5] F. Ramos et al., "IST-LASAGNE: Towards all-optical label swapping employing optical logic gates and optical flip-flops", Lightwave Technology, vol. 23, no. 10, pp. 2993-3011, 2005.